NAUKI PODSTAWOWE
CZASOPISMO TECHNICZNE
TECHNICAL TRANSACTIONS
FUNDAMENTAL SCIENCES
WYDAWNICTWO POLITECHNIKI KRAKOWSKIEJ
1-NP/2012
ZESZYT 18
ROK 109
ISSUE 18
YEAR 109

MATEUSZ BIERNACKI, KRZYSZTOF RZECKI*

URZĄDZENIE DO ZDALNEJ ADMINISTRACJI SERWEREM

REMOTE SERVER ADMINISTRATION DEVICE

Streszczenie

Do zdalnej administracji najczęściej wykorzystywane są aplikacje uruchomione na danej maszynie. W sytuacji awarii aplikacje te mogą nie zostać uruchomione, przez co administrator traci możliwość zdalnego zarządzania. Artykuł przedstawia rozwiązanie tego problemu przez wykorzystanie zewnętrz-nego urządzenia. Sprzętowe rozwiązanie jest niezależne od zarządzanej maszyny. Zastosowanie łatwo dostępnych i tanich komponentów znacznie ogranicza koszt budowy jednocześnie funkcjonalność jest porównywalna do rozwiązań komercyjnych.

Słowa kluczowe: zdalna administracja, KVM over IP, VNC

Abstract

The remote administration often used applications running on the machine that is managed. In case of failure these applications may not be run what the administrator loses the ability to remotely manage. The paper presents the solution to this problem by using an external device. The hardware solution is independent of the managed machine. Use of easily available and inexpensive components significantly reduces the cost of device while the functionality is comparable to commercial solutions.

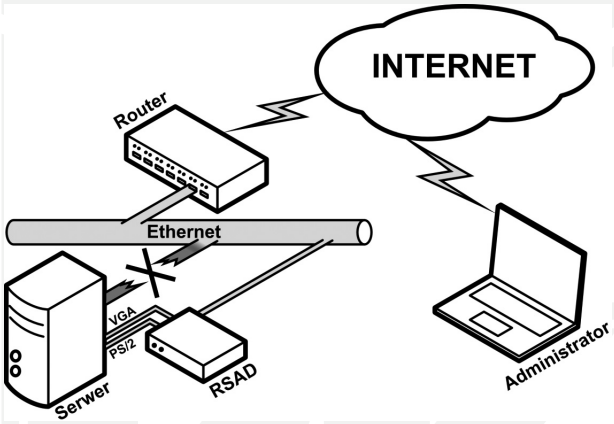
Keywords: remote administration, KVM over IP, VNC

* Inż. Mateusz Biernacki, dr inż. Krzysztof Rzecki, Instytut Teleinformatyki, Wydział Fizyki, Matematyki i Informatyki, Politechnika Krakowska.

1. Wstęp

Artykuł porusza problem polegający na braku możliwości zdiagnozowania stanu i ewentualnej awarii serwera w przypadku, gdy usługi zdalnego dostępu, takie jak SSH czy Telnet, nie odpowiadają na próby połączenia. W takiej sytuacji administrator zmuszony jest osobiście odwiedzić miejsce, w którym znajduje się dana maszyna, co często jest utrudnione albo niemożliwe w krótkim czasie. Należy również pamiętać o kosztach związanych z ewentualnym dojazdem lub karach naliczanych za przerwę w dostarczaniu usług, jakie świadczyła uszkodzona maszyna.

Rozwiązaniem powyższego problemu jest budowa urządzenia do zdalnej administracji serwerem RSAD (*Remote Server Administration Device*), które umożliwiałoby zdalne zdiagnozowanie komputera. Urządzenie takie musiałoby działać niezależnie od systemu zainstalowanego na danej maszynie. Administrator posiadałby możliwość zdalnego zalogowania się na urządzeniu i skontrolowania aktualnego stanu ekranu. Jednocześnie mógłby wprowadzać komendy za pomocą klawiatury.



Rys. 1. Zasada działania urządzenia do zdalnej administracji serwerem RSAD

Fig. 1. Principle of operations of the Remote Server Administration Device RSAD

Te podstawowe funkcje pozwoliłyby zdiagnozować problemy sprzętowe, takie jak uszkodzony dysk lub problem z kartą sieciową. Szczególnie pomocna byłaby możliwość oglądnięcia informacji pochodzących z momentu startu maszyny, w którym wspomniane usługi SSH czy Telnet nie są jeszcze uruchomione.

2. Rozwiązania komercyjne

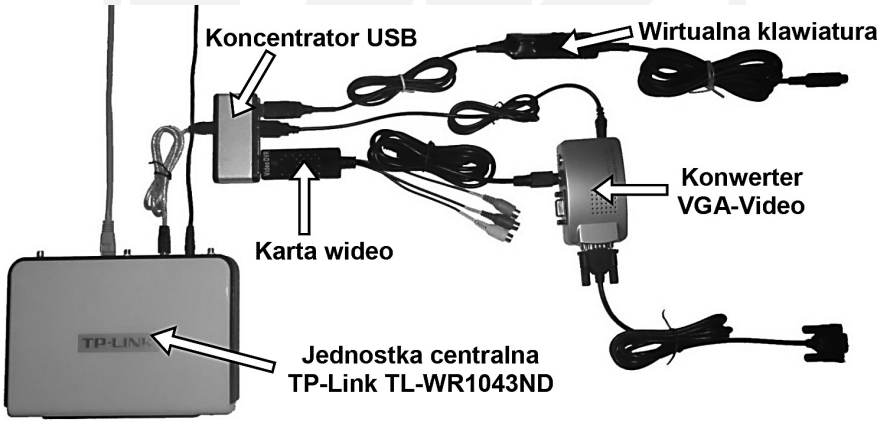
Dostępne aktualnie rozwiązania realizujące przedstawione we wstępie założenia nie są zbyt popularne, więc nie istnieje zbyt duża konkurencja wśród producentów. Pozwoliło to na sztuczne zawyżanie cen produktów, które nie są adekwatne do kosztów produkcji i zastosowanych wewnątrz technologii. W śród nich znajdują się D-Link DKVM-IP1, ATEN CN8000

i AdderLink IPEPS. Wszystkie wyposażone są w port VGA, dzięki któremu urządzenie pozyskuje obraz. Dwa porty PS/2 umożliwiają sterowanie zdalną maszyną. W zależności od modelu możliwa jest praca z rozdzielczościami nawet 1600 na 1200 pikseli. Połączenie między klientem a urządzeniem może być szyfrowane. Produkty firmy ATEN oraz AdderLink pozwalają na bezpośrednie przesyłanie plików do zdalnej maszyny przez wykorzystanie dodatkowego portu USB. Funkcja ta nosi nazwę VirtualMedia i pozwala również na instalowanie wirtualnego napędu na zdalnej maszynie, co w przypadku poważnej awarii znacznie ułatwi proces przywracania systemu operacyjnego. Model CN8000 wyposażony jest w dodatkowy port, na którym wyprowadzono sygnał PON (*Power On The Net*), umożliwiający uruchomienie zdalnej maszyny. Ceny powyższych produktów wahają się od 1500 zł do 3000 zł.

Koszt zaprojektowanego urządzenia jest znacznie niższy od ceny przedstawionych powyżej produktów komercyjnych, co stanowi jego główny atut. Za wszystkie elementy potrzebne do budowy urządzenia należy zapłacić około 400 zł. Dodatkowo w finalnym produkcie można umieścić podzespoły na jednej płycie drukowanej oraz zrezygnować z zęści dublujących się elementów, takich jak układy zasilania. Takie rozwiązanie przyczyniłoby się do zmniejszenia kosztów produkcji. Pod względem parametrów urządzenie oferuje mniejszą rozdzielczość obrazu oraz nie posiada tak wielu funkcji, jak prezentowane wyżej produkty. Jednak z jednej strony w praktyce w znakomitej większości przypadków nie są bezwzględnie wymagane, a z drugiej strony można je dodać przez modyfikację i rozbudowę oprogramowania.

3. Projekt realizacji sprzętowej

Realizację projektu podzielono na dwie części. Pierwszą z nich dotyczy budowy, a w zasadzie zestawienia sprzętu będącego w stanie wykonać założenia projektu, drugą natomiast jest oprogramowanie sprzętu i osiągnięcie pełnej funkcjonalności. Zbudowane urządzenie do zdalnej administracji serwerem RSAD w większości składa się z gotowych produktów, wyjątek stanowi moduł przesyłający sygnały sterujące (klawiatury) do komputera.



Rys. 2. Budowa urządzenia do zdalnej administracji RSAD

Fig. 2. Construction of RSAD

Urządzenie RSAD składa się z jednostki centralnej, która odpowiada za komunikację z użytkownikiem. Element ten pozyskuje informacje o obrazie oraz przekazuje sygnały sterujące. Z powodu ograniczeń w ilości portów konicznym było zastosowanie koncentratora USB. Urządzeniem odpowiedzialnym za przechwytywanie obrazu jest karta wideo na USB (Video Grabber USB). Z uwagi, iż karta ta jest w stanie odbierać sygnał jedynie w standardzie S-Video lub Composite Video, (serwery najczęściej wyposażone są w interfejs D-SUB VGA) wykorzystano sprzętowy konwerter sygnałów. Do przekazywania sygnałów sterujących zbudowano układ symulujący pracę klawiatury na PS/2. Komunikacja z układem odbywa się za pomocą portu COM, uzyskanego dzięki konwerterowi USB-COM.

3.1. Jednostka centralna

Jednostkę centralną w budowanym urządzeniu stanowi router firmy TP-Link TL-WR1043ND. Przetwarzanie obrazu jest procesem wymagającym, przez co niezbędne było wykorzystanie stosunkowo mocnego procesora i pamięci RAM o znacznej wielkości. Jeżeli zachodzi potrzeba wykorzystania większych zasobów, dużo łatwiej zarządzać nimi za pośrednictwem systemu operacyjnego. Tym sposobem ukształtowały się podstawowe wymagania stawiane jednostce centralnej. Opisane urządzenie dodatkowo musiało posiadać port USB. Jak wynika z założeń projektu konieczna jest możliwość podpięcia do sieci Ethernet, w celu komunikacji ze zdalnym klientem.

Analizując rynek, trafiono na kilka propozycji, które spełniały założone wymagania. Głównym powodem dla jakiego został wybrany router firmy TP-Link, to stosunek ceny do możliwości. TL-WR1043ND jest bezprzewodowym routerem pracującym w standardzie IEEE 802.11n, dodatkowo posiada gigabitowy przełącznik. Oparty jest na procesorze Atheros AR9132 taktowany 400MHz. Wyposażony w 32MB pamięci RAM oraz port Host USB 2.0. Dla przykładu firma Propox oferuje minimoduł MMnet1002 oparty na procesorze ARM9 taktowany częstotliwością 400Mhz, 64MB pamięci RAM z dwoma portami Host USB 2.0 oraz portem Ethernet 100Mb/s. Parametry tego modułu idealnie wpasowują się w stawiane wymagania. Cena to około 430 zł. Wykorzystany router kosztuje około 200 zł, mimo że jego parametry nieznacznie odbiegają od wspomnianego produktu MMnet1002.

Z oprogramowaniem dostarczonym przez producenta router działa jak klasyczny punkt dostępowy. Jednak od najprostszych tego typu rozwiązań, urządzenie to wyróżnia się możliwością zainstalowania otwartego oprogramowania.

Firmy Linksys, TP-Link oraz D-Link wykorzystały w swoich produktach oprogramowanie udostępnione na licencji GNU GPL, przez co były zmuszone upublicznić wszystkie zmiany w kodzie, niezbędne do poprawnej pracy swoich urządzeń. Korzystając z tych informacji, stworzono wiele otwartych wersji oprogramowania dla tych urządzeń, między innymi OpenWRT, Tomato czy DD-WRT. Oprogramowanie to poza dostarczaniem podstawowej funkcjonalności, daje przede wszystkim możliwość uruchamiania własnego kodu oraz instalowanie urządzeń innych niż przewidziane przez producenta.

Poniżej opisano krok po kroku sposób przygotowania przedstawionego routera do pracy zgodnie z postawionymi oczekiwaniami. W pierwszej kolejności należy pobrać niezbędne programy. Pierwszym z nich jest subversion, czyli system kontroli wersji. Za jego

pomocą pobrano aktualną wersje systemu OpenWRT. Na systemie posiadającym menadżer pakietów APT można zainstalować program za pomocą komendy:

```
apt-get install subversion
```

Kolejnym krokiem jest pobranie aktualnej wersje źródeł systemu OpenWRT z serwera SVN. Wykonuje się to poprzez wprowadzenie komendy:

```
svn co svn://svn.openwrt.org/openwrt/branches/backfire
```

Następnie należy wejść do nowo utworzonego katalogu backfire i uruchomić menu konfiguracji:

```
make menuconfig
```

Przed uruchomieniem, aplikacja sama sprawdza, czy nasz system posiada niezbędne narzędzia, takie jak flex czy bzip2. Brakujące programy należy pobrać za pomocą menadżera APT.

Menu konfiguracyjne to wygodna w obsłudze aplikacja o opcjach ułożonych w formie drzewa. Zadaniem użytkownika jest wybranie odpowiednich komponentów systemu OpenWRT, niezbędnych do poprawnej pracy routera. Należy pamiętać, aby ograniczać ilość komponentów ze względu na małą ilość pamięci RAM zainstalowanej w urządzeniu. Będzie ona dodatkowo potrzebna do poprawnej pracy aplikacji serwera, udostępniającego zdalnie obraz nadzorowanej maszyny. Względem podstawowej konfiguracji trzeba wybrać:

- poprawną rodzinę urządzeń – Atheros AR71xx,
- moduły odpowiedzialne za obsługę interfejsu USB: kmod-usb-core i kmod-usb2,
- moduł odpowiedzialny za wsparcie urządzeń wideo V4L: kmod-video-core,
- moduł odpowiedzialny za obsługę portu COM na USB: kmod-usb-serial-pl2303.

Niektóre z elementów ze względu na zależność od siebie same zostaną wybrane do wynikowego oprogramowania. Następnym krokiem w przygotowaniu routera do pracy jest kompilacja oprogramowania. Wykonuje się ją za pomocą komendy:

```
make
```

Jest to proces dość czasochłonny, podczas którego program sam pobiera brakujące składniki. Wstępnie skompilowane zostaną narzędzia wykorzystane później do właściwej kompilacji oprogramowania. Cały folder może zwiększyć swój rozmiar nawet do 3GB, a wynikowy plik, który zostanie wgrany na urządzenie powinien zajmować około 3MB. Oczywiście możnaby wykorzystać gotowe oprogramowanie udostępnione przez twórców OpenWRT, a następnie dograć niezbędne moduły. Jednak taki system zawiera znaczną ilość niepotrzebnych składników, które zajmują zarówno pamięć Flash (dyskową) jak i pamięć operacyjną RAM. Dodatkowym powodem, dla którego trzeba było wykonać tę czasochłonną operacje jest fakt, iż późniejsza kompilacja własnych modułów (sterownika dla urządzenia video), czy aplikacji serwera będzie korzystać z tak przygotowanych źródeł.

Po skompilowaniu należy na stronie służącej konfiguracji routera w zakładce aktualizacji oprogramowania wczytać odpowiedni plik:

```
openwrt-ar71xx-tl-wr1043nd-v1-squashfs-sysupgrade.bin
```

Producent urządzenia nie zablokował możliwości wgrywania oprogramowania pochodzącego z innego źródła. Zaznaczył przy tym, że operację taką wykonuje się na własną odpowiedzialność i wiąże się to z utratą gwarancji routera. Tak oprogramowany router pozbawiony będzie wygodnego interfejsu webowego. Jednak istnieje już możliwość zalogowania się na niego za pomocą protokołu SSH [5].

3.2. Interfejs przechwytywania obrazu VGA

Biorąc pod uwagę fakt, że urządzenie RSAD będzie współpracować z różnymi maszynami, powinno się za źródło obrazu wybrać najbardziej uniwersalne z gniazd. Takim gniazdem zapewne jest złącze monitorowe D-SUB VGA. Można je spotkać zarówno w starszych, jak i w nowych rozwiązaniach, gdyż jest to standard złącza monitorowego.

Przetwarzania obrazu bezpośrednio z portu VGA daje lepszą jakość obrazu, jednak wymaga zastosowania szybkich elementów elektronicznych. Jednocześnie przy tak szybkich przebiegach niezbędnym jest posiadanie odpowiedniego zaplecza laboratoryjnego, które pozwoliłoby na diagnozowanie ewentualnych problemów. W rezultacie istniałoby zbyt duże ryzyko niepowodzenia, dlatego postanowiono wykorzystać półprodukty w celu przechwycenia obrazu.

Tanim i łatwo dostępnym interfejsem jest się karta wideo na USB EasyCap DC60. Urządzenie jest w stanie przechwytywać obraz w standardzie S-Video i Composit Video oraz posiada popularny interfejs USB. Urządzenie przechwytuje obraz w maksymalnej rozdzielczości 720 na 576 pikseli. Wykorzystanie portu USB umożliwia podłączenie urządzenia do niemal każdego komputera. Producent zapewnia, że karta jest w stanie dostarczać 25 klatek na sekundę, jednak faktyczna ilość może zależeć od mocy obliczeniowej wykorzystanej maszyny. Z uwagi, iż produkt nie posiada interfejsu VGA niezbędne było zapewnienie odpowiedniej konwersji sygnałów. W tym celu również zastosowano komercyjny produkt. Video Converter może pracować z maksymalna rozdzielczość 1280 na 1024 piksele. Jednocześnie potrafi konwertować sygnał o bardzo niskich rozdzielczościach rzędu 640 na 480 pikseli, co pozwala zastosować je na potrzeby tego projektu. Zarówno karta wideo, jak i konwerter wyposażone są jednocześnie w oba złącza Composite Video oraz S-Video. Zdecydowano wykorzystać standard S-Video z uwagi na możliwość uzyskania lepszej jakości obrazu.

Do poprawnej pracy każdego urządzenia w systemie, niezbędnym elementem jest sterownik tego urządzenia. Producent wykorzystanej karty wideo dostarcza sterownik jedynie dla systemu Windows. W Internecie znaleziono kody źródłowe sterownika (modułu) dla systemu Linux. Po upewnieniu się, że sterownik działa na klasycznym sprzęcie należało skompilować źródła dla systemu OpenWRT. Należy pamiętać, że router wyposażony jest w procesor rodziny MIPS, przez co kompilacja wymaga zastosowania innego kompilatora. Wstępnie trzeba pobrać źródła sterownika z strony:

```
http://sourceforge.net/projects/easycapdc60/
```

Po pobraniu archiwum należy rozpakować plik do osobnego katalogu:

```
tar -xzf easycap_dc60.0.9.tar.gz
```

Kolejnym etapem jest drobna modyfikacja w plikach. Wynikają one z małej ilości dostępnej pamięci na routerze. Sterownik po podłączeniu urządzenia wstępnie alokuje 6 buforów ramki, gdzie każda z ramek może przechować maksymalny możliwy obraz. Tym sposobem alokowane jest około 6 x 1,6MB = 9,6MB. Dodatkowo alokowane są bufory dla dźwięku oraz pośrednie bufory obrazu. Łącznie sterownik alokuje ponad 16MB, co stanowi połowę dostępnej pamięci RAM. Z uwagi, iż nie będzie korzystać się z dźwięku cześć pamięci można odzyskać. Sterownik alokuje 6 buforów, jednak do poprawnej pracy wystarczą 2 z buforów, więc można zmniejszyć ich ilość. W tym celu w pliku ./src/easycap.h należy zmodyfikować następujące linie:

```
#define FIELD_BUFFER_MANY 2
#define FRAME_BUFFER_MANY 2
```

Następnym etapem jest modyfikacja skryptu ./install.sh, tak by był w stanie przygotować pliki Makefile (co jest jego głównym zadaniem) dla maszyny innej, niż na której zostanie uruchomiony. Pierwsza modyfikacja dotyczy wyłączenia obsługi dźwięku, przez co sterownik będzie wykorzystywać mniej zasobów. W tym celu należy zmodyfikować wartość zmiennej:

```
SND_USB_AUDIO=1
```

Następnie dodajemy deklaracje zmiennej MAKE_ARCH:

```
MAKE_ARCH=""if [ "x" != "x$2" ]; then
    if [ "x" != "x$3" ]; then
          MAKE_ARCH="ARCH=$2 CROSS_COMPILE=$3";
    fi
fi
```

Zmienna ta uzupełniana będzie argumentami wywołania skryptu. Teraz należy odszukać wszystkie wywołania programu make w skrypcie i do argumentów dodać zmienną MAKE_ARCH. Przykładowo linia 177, która wyglądała:

```
make -f tmpMakefile 1>./tmpsrc/tmp.out 2>./make.err
```

powinna wyglądać następująco:

```
make ${MAKE_ARCH} -f tmpMakefile 1>./tmpsrc/tmp.out 2>./make.err
```

Następnie należy wykonać skrypt z odpowiednimi parametrami. Jak już wspomniano, kompilacja oprogramowania dla routera ma jeszcze jeden cel i właśnie w tym momencie wykorzysta się narzędzia utworzone w trakcie kompilacji systemu OpenWRT. Pierwszy z argumentów to ścieżka do źródeł jądra systemu. Znajdują się one w katalogu:

```
/home/user/backfire/build_dir/linux-ar71xx/linux-2.6.32.27/
```

Kolejny argument określa architekturę, na jaką zostanie skompilowany sterownik. W tym wypadku będzie to MIPS. Ostatni argument określa prefiks, jaki zostanie dodany przed wywołaniem narzędzi do kompilacji. Dodanie tych parametrów pozwoli przygotować sterownik dla procesora w jaki wyposażony jest router. Cała instrukcja wywołująca proces konfiguracji oraz kompilacji sterownika dla karty wideo powinna wyglądać następująco:

```
./install.sh /home/user/backfire/build_dir/linux-ar71xx/linux-
2.6.32.27/mips /home/user/backfire/staging_dir/toolchain-mips_r2_
gcc-4.3.3+cs_uClibc-0.9.30.1/usr/bin/mips-openwrt-linux-uclibc-
```

Jeśli skompilowane źródła system OpenWRT oraz narzędzia znajdują się w innym miejscu niż podane w przykładzie, należy podać odpowiednie ścieżki do katalogów. Po zakończeniu procesu powinno się uzyskać plik easycap.ko w katalogu ./src/. Aby upewnić się, że jest on przeznaczony dla odpowiedniej maszyny, można wywołać polecenie file:

```
file easycap.ko
easycap.ko: ELF 32-bit MSB relocatable, MIPS, MIPS32 rel2 ver-
sion 1 (SYSV), with unknown capability 0x41000000 = 0xf676e75,
not stripped
```

Tak wygenerowany moduł należy skopiować na router oraz zainstalować za pomocą polecenia insmod:

```
insmod easycap.ko
```

W tym momencie po podpięciu karty video EasyCap w systemie powinno pojawić się urządzenie /dev/video0 [4].

3.3. Interfejs wirtualnej klawiatury

W urządzeniu do zdalnej administracji RSAD poza możliwością oglądania ekranu zdalnej maszyny niezbędne jest, by użytkownik mógł prowadzać komendy przez klawiaturę. Zazwyczaj nie ma możliwości wystawienia diagnozy usterki po samym ekranie, jaki aktualnie wyświetla maszyna. Z tego powodu koniecznym było przekazanie sterowania bezpośrednio do zdalnej maszyny. Dzięki zastosowaniu serwera x11vnc istnieje możliwość przekierowania sygnałów sterujących pochodzących z zdalnej maszyny do dowolnego programu.

Wykonano sprzętowe urządzenie, które będzie w stanie odbierać sygnały od routera, przez interfejs RS232, a następnie przekaże je do zarządzanej maszyny za pomocą portu klawiatury PS/2. O wyborze portu PS/2 zdecydował prosty algorytm transmisji oraz popularność występowania.

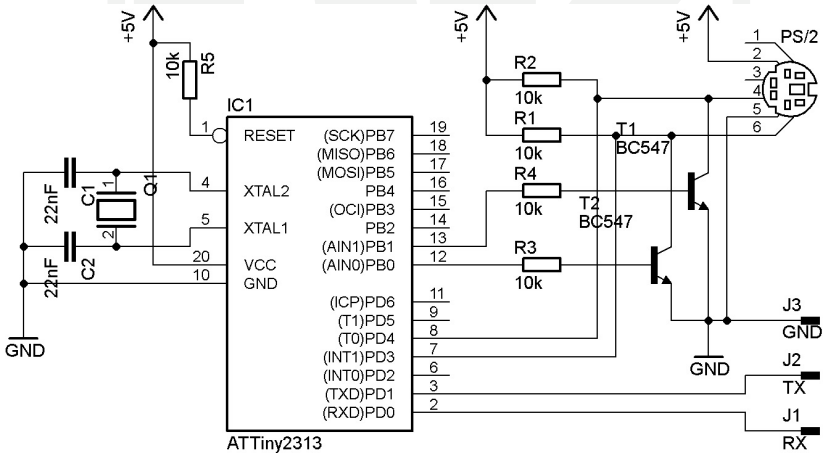
W celu przekazania sygnałów sterujących od serwera x11vnc do maszyny zbudowanego układu należało napisać prosty program parsujący dane i przesyłający je za pomocą portu COM. Program x11vnc wysyła dane w postaci:

```
Keysym 1 1 65505 Shift_L KeyPress
Keysym 1 1 75 K KeyPress
Keysym 1 0 75 K KeyRelease
Keysym 1 0 65505 Shift_L KeyRelease
```

Każda z linii odpowiada przyciśnięciu lub puszczeniu klawisza. Po otworzeniu portu COM (urządzenie /dev/ttyUSB0), każda z odebranych linii jest mapowana na kod klawisza zgodny z protokołem PS/2. Przykładowo powyżej przedstawiona sytuacja wciśnięcia klawisza „K” z użyciem klawisza Shift w protokole PS/2 jest reprezentowana jako:

```
0x12 0x42 0xF0 0x42 0xF0 0x12
```

gdzie pierwszy bit odpowiada wciśnięciu lewego klawisza Shift, następnie przesłany jest kodu klawisza K, po czym następują kody oznaczające puszczenie tych klawiszy. Dane w takiej formie po przez port COM trafiają do mikrokontrolera.



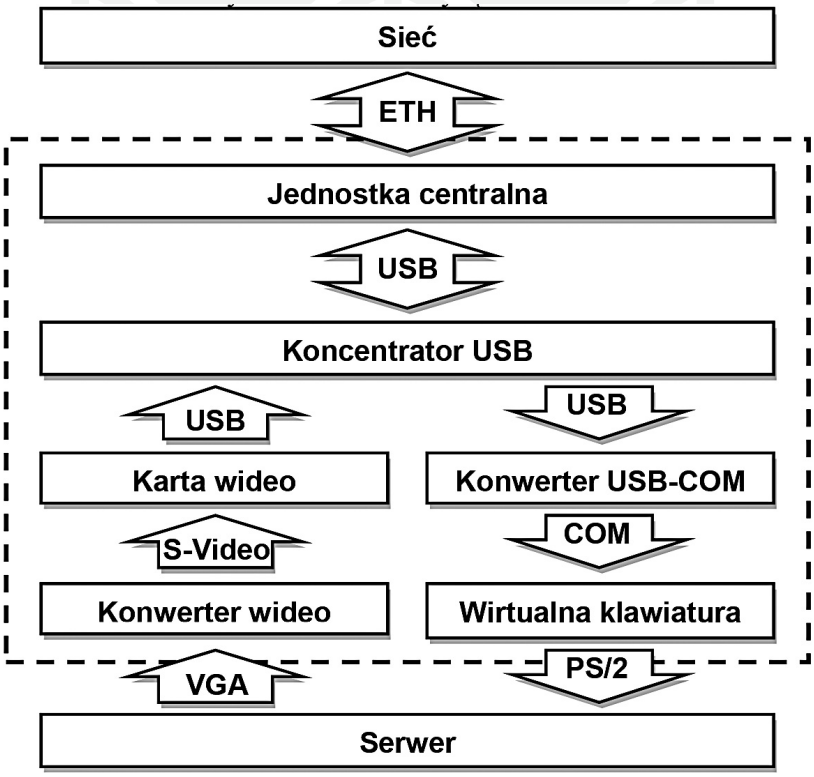
Rys. 3. Schemat układu symulującego prace klawiatury

Fig. 3. Virtual Keyboard schematic diagram

Układ symulujący prace klawiatury ma za zadanie przekazywać zdarzenia wciśnięcia lub puszczenia klawiszy do maszyny, pod którą jest podpięty. Budowa oparta jest o prosty mikrokontroler rodziny AVR ATTiny2313. Porty tego mikrokontrolera nie są w stanie pracować w układzie otwartego kolektora, co jest wymagane przy komunikacji PS/2. Z tego powodu koniecznym było zastosowanie kilku elementów zewnętrznych, takich jak tranzystory T1 oraz T2 i rezystory ograniczające prąd bazy R3 i R4. Rezystory R1 i R2 podciągają linie danych oraz linie zegarową do napięcia +5V. Układ zasilany jest bezpośrednio z portu PS/2. Za komunikacje z routerem odpowiedzialny jest konwerter USB – COM oparty na układzie Prolific PL-2303 [2].

Program mikrokontrolera został napisany w języku C. Po inicjalizacji układów peryferyjnych (zegar, przerwanie zewnętrzne INT1 oraz układy transmisji UART), główna procedura programu zostaje zapętlona. Mikrokontroler w tym stanie oczekuje nadejścia przerwania na linii INT1 lub przerwania od wewnętrznego układu UART co rozpoczyna proces przesłania znaku miedzy routerem a maszyną.

Na poniższym schemacie blokowym przedstawiono elementy składowe urządzenia do zdalnej administracji serwerem RSAD. W strzałkach zaznaczono interfejsy łączące poszczególne bloki. Kierunek strzałek obrazuje przepływ informacji.



Rys. 4. Schemat budowy urządzenia RSAD

Fig. 4. RSAD block diagram

4. Projekt realizacji oprogramowania

Podstawowym zadaniem projektowanego urządzenia jest udostępnianie zdalnemu użytkownikowi obrazu generowanego przez lokalną maszynę. Powstało kilka protokołów mających dokładnie takie przeznaczenie. Najbardziej rozpowszechnionym, a zarazem spełniającym wszystkie stawiane wymagania okazał się protokół RFB.

4.1. Protokół RFB

Protokół RFB (*remote framebuffer* – zdalny bufor ramki) jest prostym protokołem umożliwiającym zdalny dostęp do graficznego interfejsu użytkownika. Dzięki temu, że pracuje na buforze ramki, może współpracować z wszystkimi okienkowymi systemami, takimi jak X11, Windows czy Macintosh. Protokół RFB jest używany przez aplikacje typu VNC (Virtual Network Computing). Zdalna strona, gdzie znajduje się użytkownik posiadający ekran z klawiaturą i/lub myszą, jest nazywana klientem RFB. Strona, na której zachodzą zmiany w buforze ramki, czyli tam gdzie faktycznie system okienkowy się znajduje nazywana jest serwerem RFB.

Protokół RFB bazuje na koncepcji „cienkiego klienta". Przy projektowaniu kładziono nacisk, by zmniejszyć do minimum wymagania stawiane stronie klienta. Dzięki temu klient może pracować na niemal każdym sprzęcie, a implementacja aplikacji klienckich powinna być bardzo prosta.

Przekazywanie obrazu opiera się o prostą zasadę: „umieść prostokąt pikseli w danym punkcie x, y". Jest to mało efektywny, ale zarazem bardzo prosty sposób na rysowanie dowolnych składników graficznego interfejsu użytkownika. Dodając różne możliwości kompresji danych, otrzymujemy bardzo elastyczny protokół, który możemy parametryzować ze względu na przepustowość łącza, szybkość działania klienta czy serwera.

Żądanie zmiany kilku obszarów składa się na aktualizacje całego bufora ramki. Aktualizacja bufora jest przesyłana od serwera na żądanie klienta. Dzięki temu klient sam decyduje o jakości obrazu, przez częstsze lub rzadsze żądania aktualizacji. Pozwala to na ograniczenie ilość transmitowanych danych, w zależności od przepustowości połączenia.

Część protokołu odpowiedzialna za przekazywanie sygnałów sterujących bazuje na klasycznej stacji roboczej z klawiaturą oraz myszą. Zdarzenia sygnałów wejściowych są przesyłane od klienta do serwera za każdym razem, gdy użytkownik wciśnie przycisk na klawiaturze, myszce lub przesunie wskaźnik.

Od kodowania zależy, w jaki sposób dane o aktualizowanym obszarze zostaną przesłane do klienta. Każdy obszar jest poprzedzony nagłówkiem zawierającym informacje o położeniu prostokąta, jego rozmiarze oraz sposobie kodowanie danych o pikselach. Sposoby kodowania aktualnie zdefiniowane to: Raw, CopyRect, RRE, Hextile oraz ZRLE. W praktyce stosowane są ZRLE, Hextile i CopyRect, ponieważ zapewniają najlepszy stosunek kompresji dla przeciętnego pulpitu. Szczegółowe opisy sposobów kodowania danych o pikselach są zawarte w specyfikacji protokołu RBF.

4.2. Serwer x11vnc

Serwer x11vnc jest jednym z wielu implementacji serwera VNC. Został wybrany z kilku powodów. Podstawowym atutem jest możliwość pracy z urządzeniami wideo, a nie bezpośrednio na buforze ekranu maszyny, na którym został uruchomiony. W szczególności ta

własność wyróżnia serwer x11vnc od innych implementacji serwerów VNC. Dodatkową zaletą jest możliwość przekazania sygnałów sterujących do dowolnej aplikacji. W klasycznym rozwiązaniu zdarzenia przyciśnięcia klawiszy pochodzące od klienta VNC trafiają do aplikacji serwera, a następnie do systemu, na którym serwer jest uruchomiony. W przypadku urządzenia do zdalnej administracji RSAD, zdarzenia te muszą trafiać do innej maszyny niż ta na której uruchomiono serwer VNC.

Wstępnie serwer x11vnc był testowany na klasycznym PC. Okazało się, że opisana przez twórców możliwość przechwytywania obrazu z urządzeń video nie jest w pełni funkcjonalna. Wspiera ona jedynie urządzenia pracujące w standardzie Video4Linux w wersji 1. Aktualnie stosowana wersja 2 nie jest w pełni kompatybilna ze swoim poprzednikiem. Z tego powodu wynikły problemy z użyciem karty video EasyCap. Kod serwera x11vnc jest publicznie dostępny więc należało zmodyfikować źródła tak, by działały z urządzeniami video w wersji V4L2.

Kod nie został napisany zgodnie ze standardami. Występuje wiele zmiennych globalnych oraz niewiele komentarzy. Utrudniło to wprowadzenie niezbędnych zmian. Pierwsza modyfikacja dotyczyła dodania możliwości zmiany parametrów obrazu. Konieczne było wywołanie kilku funkcji ioctl(), które odpowiadały za zmianę parametrów. Kolejne zmiany dotyczyły sposobu odczytu danych. W oryginalnym kodzie twórcy wykorzystali funkcje read() w celu pobrania obrazu z urządzenia. Niestety sterownik urządzenia EasyCap nie wspierał trybu IO_METHOD_READ, co nie było sprawdzane przez program. Sterownik mógł pracować w trybie IO_METHOD_MMAP więc niezbędnym było zastosowanie funkcji mmap(). Modyfikacje dotyczyły dodatkowo pliku screen.c, w którym dodano żądanie udostępnienia buforu [3].

Kiedy okazało się, że po tych modyfikacjach program działa, należało skompilować go dla procesora dostępnego w routerze. Wykonano konfiguracje przez wywołanie skryptu ./configure dodając odpowiednie argumenty:

```
CC=/home/user/backfire/staging_dir/toolchain-mips_r2_gcc--4.3.3+cs_uClibc-0.9.30.1/usr/bin/mips-openwrt-linux-gcc ./configure --host=mips --without-avahi --without-x --without-macosx-native --without-uinput --without-fbdev --without-dpms --without-fbpm --without-xkeyboard --without-xinerama --without-xrandr --without-xfixes --without-xdamage --without-xtrap --without-xrecord --with-zlib=/home/user/backfire/staging_dir/target-mips_r2_uClibc-0.9.30.1/usr
```

Pierwszy element określa kompilator. Następnie dodano opcje wyłączające zbędne funkcjonalności serwera, dzięki czemu zmniejszono rozmiar pliku wykonywalnego. Po tak przeprowadzonej konfiguracji można było uruchomić kompilacje:

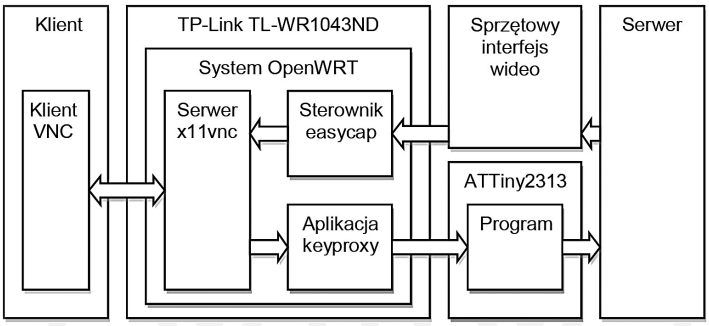
```
make
```

Po przeniesieniu programu na router należało go uruchomić podając odpowiednie parametry. Najważniejszym krokiem było wybranie źródła obrazu. W tym wypadku jest to urządzenie /dev/video0, a następnie należało odpowiednio sparametryzować to urządzenie, ustalając rozdzielczość, odpowiednie wejście sygnału czy format kolorów. Ostatnim argumentem jest żądanie przekazania sygnałów sterujących do własnej aplikacji, odpowiedzialnej za wysyłanie odpowiednich kodów klawiszy na port COM [6].

```
./x11vnc -rawfb video:w=720,h=576,bpp=32,inp=5@720x576x32:ff000000/ff0000/ff00 -many -shared -pipeinput "./keyproxy /dev/ttyUSB0 19200"
```

4.3. Klient

Użytkownik komunikuje się z urządzeniem RSAD za pomocą aplikacji klienckiej. Dzięki zastosowaniu popularnego protokołu RFB istnieje możliwość skorzystania z wielu gotowych aplikacji. Wykorzystując urządzenie typu smartphone, na które powstało wiele klientów VNC, dostęp do zdalnej maszyny można mieć w każdym miejscu i czasie. W celu ułatwienia połączenia, można skorzystać z apletu JAVA, który automatycznie zostanie uruchomiony po wywołaniu w przeglądarce adresu urządzenia. Takie rozwiązanie jest stosowane przez producentów komercyjnych produktów, przez co nie wymusza na użytkownikach instalowania dodatkowych aplikacji.



Rys. 5. Schemat przepływu danych pomiędzy aplikacjami

Fig. 5. Scheme of data flow between applications

5. Możliwości rozbudowy

Biorąc pod uwagę rozbudowę zaprezentowanego rozwiązania, można by pomyśleć nad dodaniem możliwości sterowania myszą. Dodatkowa możliwość sterowania dostępem do zasilania pozwoliłaby na natychmiastowe odłączenie maszyny. Z uwagi, iż forma przechwytywania obrazu nie pozwala na osiągnięcie większych rozdzielczości, można byłoby zbudować interfejs obrazu, bazujący bezpośrednio na porcie VGA bez konwertowania do formatu S-Video. Pożyteczną funkcjonalnością może okazać się opcja przesyłania danych przez port USB do zdalnej maszyny. Wykorzystując możliwość uruchomienia dowolnej aplikacji na systemie OpenWRT, można wykorzystać urządzenie RSAD do dodatkowych celów, np. takich jak alarmowanie drogą mailową czy SMS o ewentualnej awarii serwera. Instalując prosty serwer WWW, możemy mieć wgląd na warunki panujące w serwerowni, np. temperatura i wilgotność.

6. Wnioski

Urządzenie posiada podstawową funkcjonalność komercyjnych produktów. Jednocześnie koszt budowy jest około pięciokrotnie mniejszy od ceny tychże produktów. Atutem urządzenia RSAD jest również możliwość dodatkowego wykorzystania centralnej jednost-

ki, którą stanowi router. Dzięki zastosowaniu przy budowie urządzenia modułów, takich jak karta wideo na USB, koncentrator USB czy konwerter VGA – Video, znacznie ułatwiono proces projektowania urządzenia. Może ono posłużyć jako prototyp do budowy komercyjnego odpowiednika. Urządzenie ostateczne składałoby się z układów elektronicznych poszczególnych modułów, przez co koszt budowy byłby dodatkowo niższy. W porównaniu z urządzeniami tego typu istniejącymi na rynku zaproponowane rozwiązanie nie posiada funkcji sterowania myszą. Jednak istnieje możliwość rozbudowy urządzenia RSAD przez wykorzystanie wolnych portów koncentratora USB.

Zastosowane w projekcie aplikacje wymagały modyfikacji z uwagi na fakt, iż zostały zaprojektowane do pracy na sprzęcie o znacząco większych możliwościach niż docelowy router. Ze względu na różnice w architekturach kompilacja wymagała zastosowania odmiennych narzędzi niż aplikacja przeznaczona na klasyczną maszynę typu desktop.

Artykuł zgłoszony do publikacji w roku 2010.

Literatura

[1] Metzger P., Jełowicki A., *Anatomia PC*, Wyd. IV, Helion, Gliwice 1998.
[2] Hadam P., *Projektowanie systemów mikroprocesorowych*, Wydawnictwo BTC, Warszawa 2004.
[3] *Video for Linux Two API Specification* (http://v4l2spec.bytesex.org).
[4] *OpenWrt non-standard module compiling* (http://www.farlock.org/nslu2/openwrt-non-standard-module-compiling).
[5] *OpenWrt Wiki Documetation* (http://wiki.openwrt.org).
[6] *x11vnc Documentation* (http://www.karlrunge.com/x11vnc).
[7] *The PS/2 Protocol* (http://www.computer-engineering.org/ps2protocol).